Numerical Analysis and CUDA-based Optimization of the Wave Equation

SUN LISHUANG
Supervisor： Mike Peardon, Kirk M. Soodhalter
MSc for High-performance Computing sinl1@tcd.ie

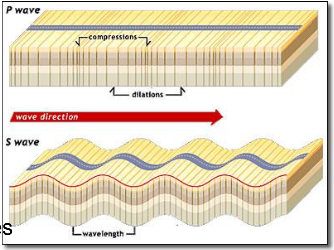
2025.09.08

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Content

Motivation & Background

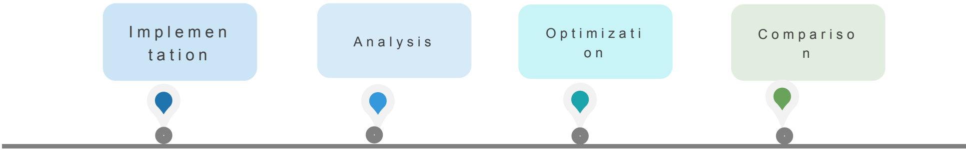
- Wave equation: Fundamental in applied mathematics and physics (seismology, acoustics, electromagnetics).
- Limitation: Analytical solutions only for simple cases; numerical simulations needed for realistic problems.
- HPC Need: High computational demand, especially for fine grids or higher dimensions.
- Compare CPU and GPU in terms of accuracy and performance



https://www.sms-tsunami-warning.com/pages/seismic-waves

Project Objectives

- Implement the Leapfrog finite difference method for 1D and 2D.
- Analyze stability and convergence properties.
- Port and optimize the solver on GPU using CUDA.
- Compare CPU and GPU implementations in terms of accuracy and performance.



Finite Difference Method Selection

Scheme	Accuracy	Stability	Numerical Behavior	HPC Suitability
Lax	First-order in time, second-order in space	CFL: $\lambda \leq 1$	Strong numerical dissipation	High (simple stencil)
Leapfrog	Second-order in time and space	CFL: $\lambda \leq 1$	No dissipation, parasitic mode possible	Very high (efficient stencil)
Lax–Wendroff	Second-order in time and space	CFL: $\lambda \leq 1$	Dispersive oscillations near discontinuities	High (slightly more complex stencil)
Crank–Nicolson	Second-order in time and space	Unconditionally stable	No dissipation, implicit solver required	Moderate (linear solve overhead)

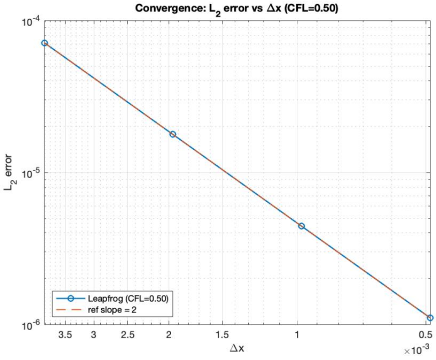
- leapfrog:

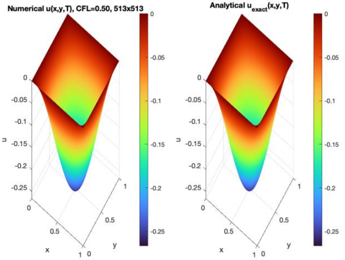
$$u_j^{n+1} = 2u_j^n - u_j^{n-1} + \lambda^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n) \quad , n>1,\ 1 \leq j \leq N-2$$

Chanllenges & Solutions

- Selecting Numerical Methods
- Code Debugging
- Technical Difficulties in HPC Optimization
- Encountering Superconvergence During Data Analysis

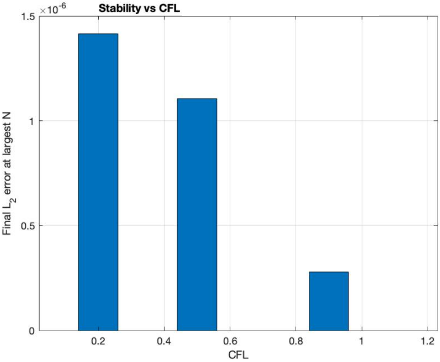
Results--Accuracy Verification （1D & 2D)

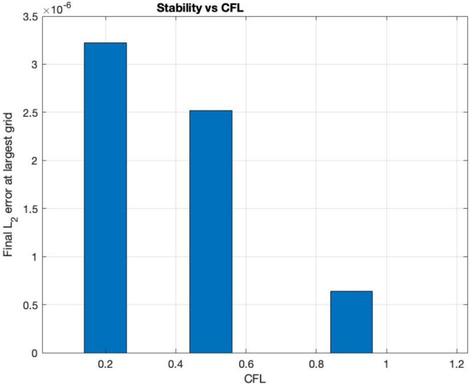




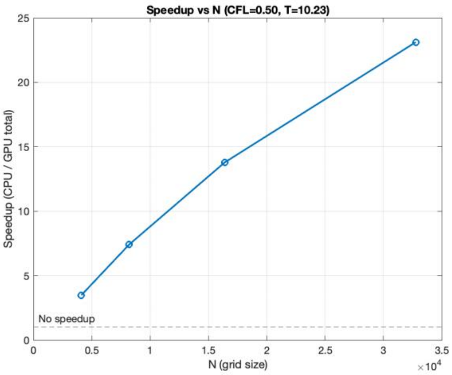
(b) Numerical vs. analytical surfaces at $t = T$.

Results--Stability Verification: Obeys CFL Rule (1D &2D)

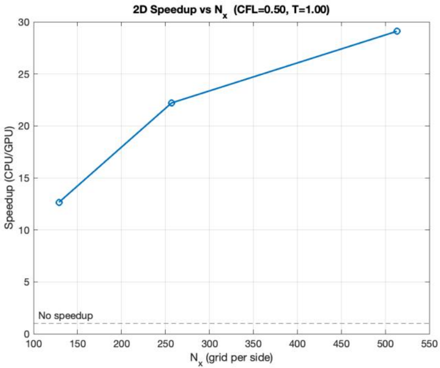




Results_Speedup(1D & 2D)



Fif 5.8



Fif 5.16b

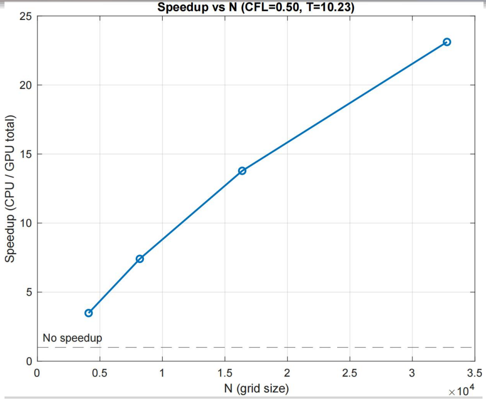
What do these results mean?

- Memory Is the Key Bottleneck
- Shared Memory + Boundary Fusion Are Optimization Core
- GPUs Excel at Large-Scale Problems

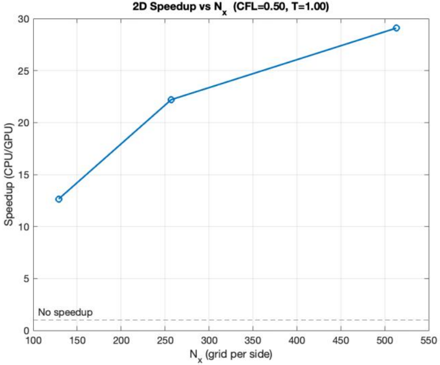
Q & A

Thank you all for listening.

Results _ Speedup(2D)



(a) Baseline CUDA speedup vs. N_x



(b) Optimized CUDA speedup vs. N_x.

Numerical Method

- 1D:

first step: $u_j^1 = u_j^0 + \Delta t\, g(x_j) + \frac{1}{2} \lambda^2 (u_{j+1}^0 - 2u_j^0 + u_{j-1}^0)$ *, g=0*

leap-frog: $u_j^{n+1} = 2u_j^n - u_j^{n-1} + \lambda^2 (u_{j+1}^n - 2u_j^n + u_{j-1}^n)$ *, n> 1, 1≤j≤N-2*

- 2D:

first step: $u_{i,j}^1 = u_{i,j}^0 + \frac{1}{2} (c\Delta t)^2\, \Delta_h u_{i,j}^0$ *, on, 1≤ i ≤ N_x-2, 1≤ j ≤ N_y-2*

leap-frog: $u_{i,j}^{n+1} = 2u_{i,j}^n - u_{i,j}^{n-1} + (c\Delta t)^2\, \Delta_h u_{i,j}^n$